

App. No. 10/001,521  
Amd. dated August 10, 2006  
Reply to Final Office Action of February 10, 2006

**Amendments to the Claims:**

This listing of claims will replace all prior versions and listings of claims in the application:

**Listing of Claims:**

1. (Currently Amended): A method for conducting an exchange of data with a terminal-based application program comprising:
  - (a) mapping a plurality of available states within a terminal data stream of the terminal-based application program to respective discrete state definitions within a finite state machine, including:
    - (1) interpreting any element, terminal command, data item, or sequence of terminal commands and data items within the terminal data stream as a discrete state having a respective one of the state definitions, and
    - (2) using an object model containing a set of interfaces, said interfaces being utilized as a basis for the state definitions;
  - (b) prompting a user for rules criteria and expected outcomes of the respective state definitions;
  - (c) aggregating the plurality of available states within the terminal data stream to eliminate redundant states, including:
    - (1) accessing data in the plurality of available states;
    - (2) unifying and resolving the data in the plurality of available states to eliminate the redundant states; and,
  - (d) defining a plurality of state transition rules which are utilized to manipulate the state definitions within the finite state machine.
2. (Original): The method of claim 1, wherein the object model contains the set of interfaces and a set of classes.

Appl. No. 10/001,521  
Amdt. dated August 10, 2006  
Reply to Office action of June 2, 2005

1 3. (Original): The method of claim 1, wherein manipulation of the state transition  
2 rules allows for two-way communication between a client and the terminal-based  
3 application program.

1 4. (Original): The method of claim 1, further comprising creating one or more data  
2 sets, each comprising a definable set of data elements from within the terminal-based  
3 application program.

1 5. (Original): The method of claim 1, wherein the object model is a distributed  
2 object transaction model, which allows for access to data from the terminal-based  
3 application program by any local or remote client service, system, or application.

1 6. (Previously Presented): The method of claim 1 further comprising unifying and  
2 resolving multiple terminal-based applications through unification and resolution of a  
3 plurality of instances of a program that performs steps (a) and (d).

1 7. (Previously Presented): The method of claim 6, further comprising creating a  
2 plurality of data sets, each data set being formed through a unification of multiple data  
3 sets from the plurality of instances of the program that performs steps (a) and (d).

1 8. (Original): The method of claim 6, wherein each object model is a distributed  
2 object transaction model which allows for access to data from the terminal-based  
3 application program by any local or remote client service, system, or application.

1 9. (Previously Presented): The method of claim 6, further comprising using an  
2 expert system which accesses and resolves data items from the plurality of instances of  
3 the program that performs steps (a) and (d) and translates them into cohesive super sets of  
4 data.

1 10. (Previously Presented): The method of claim 9, further comprising using the  
2 object model for creating data resolution and translation rules.

Appl. No. 10/001,521  
Amdt. dated August 10, 2006  
Reply to Office action of June 2, 2005

1 11. (Previously Presented): The method of claim 9 further comprising using the  
2 object model for definition of actions to proactively resolve data errors or discrepancies  
3 across an underlying plurality of instances of the program that performs steps (a) and (d).

1 12. (Previously Presented): The method of claim 1, further comprising altering an  
2 interface presented to a user of the terminal-based application program through addition  
3 of one of a group consisting of new screens and new data fields within existing screens,  
4 wherein the new screens and new data fields are populated with data retrieved from an  
5 alternate data source.

1 13. (Original): The method of claim 12, further comprising monitoring the terminal  
2 data stream.

1 14. (Original): The method of claim 13, further comprising:  
2 recognizing pre-defined states within the terminal data stream, which define one  
3 or more states during a user's interaction with the terminal-based application program;  
4 and  
5 presenting the new data screens or fields to the user.

1 15. (Previously Presented): The method of claim 12, wherein the object model  
2 describes interaction between the alternate data source and a program that performs steps  
3 (a) and (d).

1 16. (Original): The method of claim 12, wherein the object model describes:  
2 the addition of new user screens or the addition of new data fields to existing  
3 application screens, and  
4 interaction between the user and the new screens or fields.

Appl. No. 10/001,521  
Amdt. dated August 10, 2006  
Reply to Office action of June 2, 2005

1 17. (Original): The method of claim 1 further comprising using software tools to  
2 automate creation and maintenance of an integration system based on knowledge of a  
3 domain of the terminal-based application program.

1 18. (Previously Presented): A system for conducting an exchange of data with a  
2 terminal-based application program comprising:  
3 a finite state machine, in which a plurality of available states within a terminal  
4 data stream of the terminal-based application program are mapped to respective discrete  
5 state definitions, including:

6 means for interpreting any element, terminal command, data item, or  
7 sequence of terminal commands and data items within the terminal data stream as  
8 a discrete state having a respective one of the state definitions;

9 means for using an object model containing a set of interfaces, said  
10 interfaces being utilized as a basis for the state definitions;

11 means for prompting a user for rules criteria and expected outcomes of the  
12 respective means for aggregating the plurality of available states within the  
13 terminal data stream to eliminate redundant states;

14 and

15 means for defining a plurality of state transition rules which are utilized to  
16 manipulate the state definitions within the finite state machine.

1 19. (Original): The system of claim 18, wherein the object model contains the set of  
2 interfaces and a set of classes.

1 20. (Canceled)

1 21. (Canceled)

1 22. (Canceled)

Appl. No. 10/001,521  
Arndt. dated August 10, 2006  
Reply to Office action of June 2, 2005

1 23. (Original): The system of claim 18 wherein multiple terminal-based applications  
2 are unified and resolved through unification and resolution of a plurality of instances of  
3 the finite state machine.

1 24. (Original): The system of claim 23, further comprising a plurality of data sets,  
2 each data set being formed through a unification of multiple data sets from the plurality  
3 of instances of the finite state machine.

1 25. (Original): The system of claim 23, wherein each object model is a distributed  
2 object transaction model which allows for access to data from the terminal-based  
3 application program by any local or remote client service, system, or application.

1 26. (Original): The system of claim 23, further comprising an expert system which  
2 accesses and resolves data items from the plurality of instances of the finite state machine  
3 and translates them into cohesive super sets of data.

1 27. (Previously Presented): The system of claim 26, wherein the object model is used  
2 for creating data resolution and translation rules.

1 28. (Previously Presented): The system of claim 26 wherein the object model is used  
2 for definition of actions to proactively resolve data errors or discrepancies across an  
3 underlying plurality of instances of the finite state machine.

1 29. (Previously Presented): The system of claim 18, further comprising an interface  
2 presented to a user of the terminal-based application program, the interface formed  
3 through addition of one of a group consisting of new screens and new data fields within  
4 existing screens, wherein the new screens and new data fields are populated with data  
5 retrieved from an alternate data source.

Appl. No. 10/001,521  
Amdt. dated August 10, 2006  
Reply to Office action of June 2, 2005

1 30. (Original): The system of claim 29, further comprising means for monitoring the  
2 terminal data stream.

1 31. (Original): The system of claim 30, wherein:  
2 pre-defined states are recognized within the terminal data stream, which define  
3 one or more states during a user's interaction with the terminal-based application  
4 program; and  
5 the new data screens or fields are presented to the user.

1 32. (Original): The system of claim 29, wherein the object model describes  
2 interaction between the alternate data source and the finite state machine.

1 33. (Original): The system of claim 29, wherein the object model describes:  
2 the addition of new user screens or the addition of new data fields to existing  
3 application screens, and  
4 interaction between the user and the new screens or fields.

1 34. (Original): The system of claim 18 further comprising software tools that  
2 automate creation and maintenance of an integration system based on knowledge of a  
3 domain of the terminal-based application program.

1 35. (Previously Presented): The system of claim 18, further comprising a tool which  
2 automates capture of a terminal data stream and the creation of state definitions for a  
3 particular terminal-based application.

1 36. (Original): The system of claim 35, wherein the tool allows the user to define the  
2 data sets which will be made available.

Appl. No. 10/001,521  
Amdt. dated August 10, 2006  
Reply to Office action of June 2, 2005

1 37. (Original): The system of claim 36, wherein the tool allows the user to define  
2 state transition rules to access and manipulate the data sets, to read and write data  
3 elements, using a point-and-click flowchart-style interface.

1 38. (Original): The system of claim 35, further comprising software tools which  
2 automate creation and maintenance of a unification and resolution system based on a  
3 knowledge of underlying integration systems being unified.

1 39. (Original): The system of claim 35, wherein the tool allows the user to define the  
2 data super-sets which will be made available by the system.

1 40. (Original): The system of claim 35, wherein the tool allows the user to create and  
2 define data unification and resolution rules.

1 41. (Previously Presented): The system of claim 35, wherein the tool allows the user  
2 to define rules to manage data errors and discrepancies in the terminal data stream.

1 42. (Previously Presented): A computer readable medium encoded with computer  
2 program code, wherein when the computer program code is executed by a processor, the  
3 processor performs a method for conducting an exchange of data with a terminal-based  
4 application program comprising:

5 (a) mapping a plurality of available states within a terminal data stream of the  
6 terminal-based application program to respective discrete state definitions within a finite  
7 state machine, including:

8 (1) interpreting any element, terminal command, data item, or sequence of  
9 terminal commands and data items within the terminal data stream as a discrete  
10 state having a respective one of the state definitions, and

11 (2) using an object model containing a set of interfaces, said interfaces being  
12 utilized as the basis for the state definitions;

Appl. No. 10/001,521  
Amdt. dated August 10, 2006  
Reply to Office action of June 2, 2005

- 13 (b) prompting a user for rules criteria and expected outcomes of the respective state  
14 definitions;  
15 (c) aggregating the plurality of available states within the terminal data stream to  
16 eliminate redundant states; and,  
17 (d) defining a plurality of state transition rules which are utilized to manipulate the  
18 state definitions within the finite state machine.

1 43. (Previously Presented): The computer readable medium of claim 41, wherein the  
2 method further comprises unifying and resolving multiple terminal-based applications  
3 through unification and resolution of a plurality of instances of a program that performs  
4 steps (a) and (d).

1 44. (Previously Presented): The computer readable medium of claim 41, wherein the  
2 method further comprises using software tools to automate creation and maintenance of  
3 an integration system based on knowledge of a domain of the terminal-based application  
4 program.

- 1 45. (Previously Presented) A method of defining and configuring the exchange of  
2 data with a terminal-based application, comprising:  
3 (a) accessing a targeted legacy application on at least one legacy application server;  
4 (b) evaluating operated information and defined rules on the targeted legacy  
5 application;  
6 (c) modeling the targeted legacy application with a legacy application profile, screen,  
7 and data stream definitions;  
8 (d) providing automated and dynamically directed execution and runtime processing  
9 environment operating concurrently and coordinated across the at least one legacy  
10 application server;  
11 (e) providing processed legacy information objects using the objectification  
12 definitions in customizable formats and structures for access by multiple requesting  
13 applications.



Express Mail Label No. EL554221321US

R0090-00001

- 1 46. (New) The method of claim 14, wherein the unifying and resolving of the data in  
2 the plurality of available states to eliminate the redundant states further comprises:  
3 synchronizing the data in the plurality of available states; and  
4 correcting spelling errors or data entry errors in the data in the plurality of  
5 available states.